2023 전기 졸업과제 중간보고서

블록체인 기반 부산 지역 기부 플랫폼

팀명: Chaining

분과명: D

팀장: 201970146 이아영

팀원: 201724420 김동찬

201624581 정진규

담당교수: 권동현

목차

1. 과제 개요

1-1. 과제 배경

기부는 사회적으로 매우 중요한 역할을 한다. 그러나 국내에서는 기부문화가 미처 활성화되어 있지 않아 기부율이 감소하고 있다. 통계청에서 2019년도에 실시한 조사에 따르면 기부 경험이 없는 사람은 74.4%로, 기부 경험이 있는 사람(25.6%)의 약 3배였다.

이러한 문제를 해결하고자, 부산 지역 내에서 기부를 할 수 있는 플랫폼을 만들어 블록체인 기술과 스마트 컨트랙트를 활용하여 기부의 투명성과 신뢰성을 보장하고자 한다. 이를 위해, 이더리움 Dapp 플랫폼을 개발하고, 기부 증서를 NFT를 활용한 토큰으로 발행하며, 기부 횟수에 따라 기부 등급을 정하여 기부 문화를 활성화하고자 한다.

1-2. 과제 목표

본 과제에서는 부산 지역 내 기부를 할 수 있는 이더리움 Dapp 플랫폼을 개발함으로써 블록체인 기술 및 스마트 컨트랙트를 활용하여 기부의 투명성과 신뢰성을 보장하고자 한다.

기부 증서를 NFT를 활용한 토큰으로 발행하여 기부자의 기부 내역을 증명하고, 기부 횟수에 따라 기부 등급을 정하여 활동적인 기부 문화를 조성하도록 한다. 또한 기부 프로필 생성 및 공유가 가능한 시스템을 구현하여 다른 사용자와 소통할 수 있는 기능을 제공한다.

2. 설계 변경 내역

2-1. 플로우차트 변경

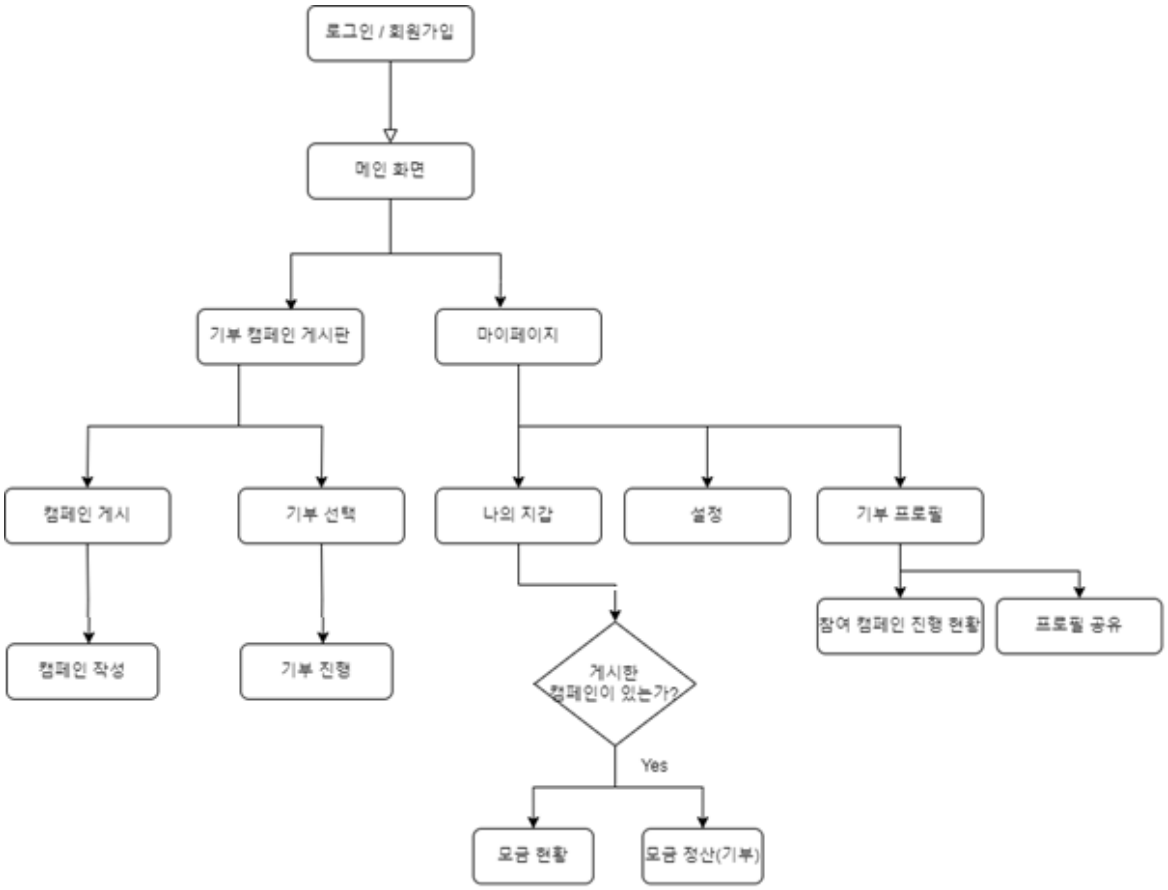


그림 1. 플로우차트

사용하는 MetaMask의 특성 및 보안 등을 고려하여 기존의 '지갑 충전 및 지갑 생성' 기능을 제외하여 개발하기로 변경되었다.

2-2. 설계 구조 변경

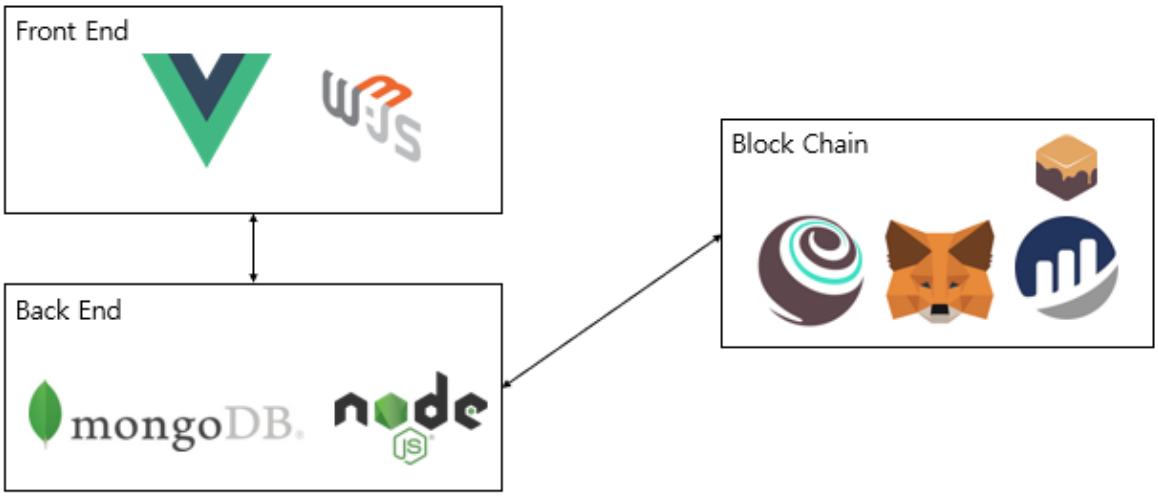


그림 2. 설계에 이용될 프레임워크

node.js와의 연동성을 고려하여 데이터베이스를 MongoDB로 변경되었다.

3. 과제 진행도

3-1. 구성원별 진척도

이아영	스마트 컨트랙트 설계 Web3.js로 연동 진행
김동찬	HTTP 메소드 등을 이용하여 게시판 기능 설계 Postman을 이용하여 REST API Pagination 설계
정진규	Vue를 이용하여 게시판 설계

3-2. 중간 결과 보고

1) 블록체인

```
//주소.balance
//주소의 현재 갖고 있는 이더의 잔액을 의미함.  //  msg.value는 송금액

// msg.sender
// msg.sender는 스마트컨트랙을 사용하는 주체

contract Donation{
    event SendInfo(address _msgSender, uint256 _currentValue);
    event MyCurrentValue(address _msgSender,uint256 _value);
    event CurrentValueOfSomeone(address _msgSender, address _to, uint256 _value);


   //이더를 보낼라면
    function sendEther(address payable _to) public payable {
        require (msg.sender.balance >= msg.value , "your balance is not enough");
        //트랜스퍼 함수를 통해 이더 송금
        _to.transfer(msg.value);
        //보낸 나의 주소, 내 잔고
        emit SendInfo(msg.sender, (msg.sender).balance);
    }
    //그냥 현재 잔고확인 함수
    function checkValueNow() public{
        emit MyCurrentValue(msg.sender, (msg.sender).balance);
    }
    //어떤 계정의 잔고 확인하고 싶을때, 확인하고 싶은 주소를 매개변수로 넣기
    function checkUserValue(address _to) public{
        emit CurrentValueOfSomeone(msg.sender, _to, _to.balance);
    }

}
```

그림 3. 스마트 컨트랙트 구현

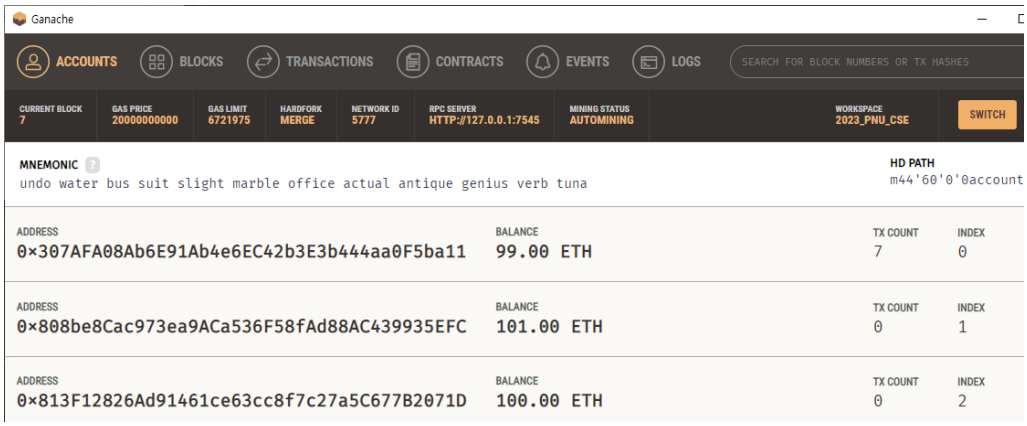


그림 4. Ganache를 이용한 블록체인 테스트넷

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <script type="text/javascript" src="./node_modules/web3/dist/web3.min.js"></script>
    <script type="text/javascript" src="./views/cont.js"></script>
    <script type="text/javascript">
        window.addEventListener('load', function () {
            if (typeof web3 !== 'undefined') {
                console.log('Web3 Detected! ' + web3.currentProvider.constructor.name)
                window.web3 = new Web3(web3.currentProvider);
            } else {
                console.log('No Web3 Detected... using HTTP Provider')
                window.web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:7545/"));
            }



            var address, wei, balance;
            address = document.getElementById("address").innerHTML;

            var balance = web3.eth.getBalance(address);
                const promise = balance;
                const getData = () => {
                promise.then((dummyData) => {
                    document.getElementById("output").innerHTML = dummyData + " ETH";
                });
                };
                getData();
        })
    </script>
</head>
<body>
    <h1>Chaining Mypage</h1>
    <p>나의 주소:</p>
    <p id="address">0x307AFA08Ab6E91Ab4e6EC42b3E3b444aa0F5ba11</p> <!--추후 db에서 받아오도록 수정-->
    <p>현재 금액:</p>
    <div id="output"></div>
</body>
</html>
```

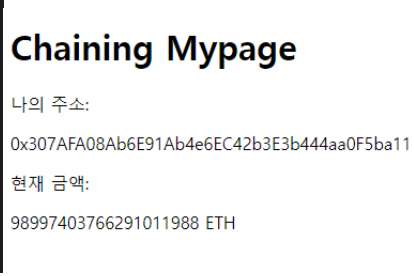


그림 5. Web3.js를 이용하여 블록체인 연동 구현

```
window.addEventListener('load', function () {
    if (typeof web3 !== 'undefined') {
                console.log('Web3 Detected! ' + web3.currentProvider.constructor.name)
                window.web3 = new Web3(web3.currentProvider);
            } else {
                console.log('No Web3 Detected... using HTTP Provider')
                window.web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:7545/"));
            }


    //contract abi 설정
    const ABI = [ ···
  //window.Buffer = require('buffer/').Buffer;
    //배포된 컨트랙트 주소 설정
        var contractAddress = "0x90199175d711F04515B06AD66fB0c62e9bB0284B";

        var MyContract = new web3.eth.Contract(ABI, contractAddress);

      var from_account = '0x307AFA08Ab6E91Ab4e6EC42b3E3b444aa0F5ba11';
      var to_account = '0x808be8Cac973ea9ACa536F58fAd88AC439935EFC';
      const privateKey = '0x58c1fc1f1ae6b4e215166b233a1d1ac84cd82d48bb2cc082711cdf3f2aa14d61';

      var userAccount = web3.eth.accounts[0]

      var accountInterval = setInterval(function() {
        // 계정이 바뀌었는지 확인
        if (web3.eth.accounts[0] !== userAccount) {
            userAccount = web3.eth.accounts[0];
            // 새 계정에 대한 UI로 업데이트하기 위한 함수 호출
            updateInterface();
        }
        }, 100);

        let deploy = MyContract.deploy({
                    data: contractAddress,
                    from: from_account}).encodeABI();
```

그림 6. Web3.js 코드

- 현재 web3.js를 이용하여 블록체인과 프론트엔드 사이 연동 진행 중
- 연동 작업 후 etherScan API 작업 예정

2) Front-end

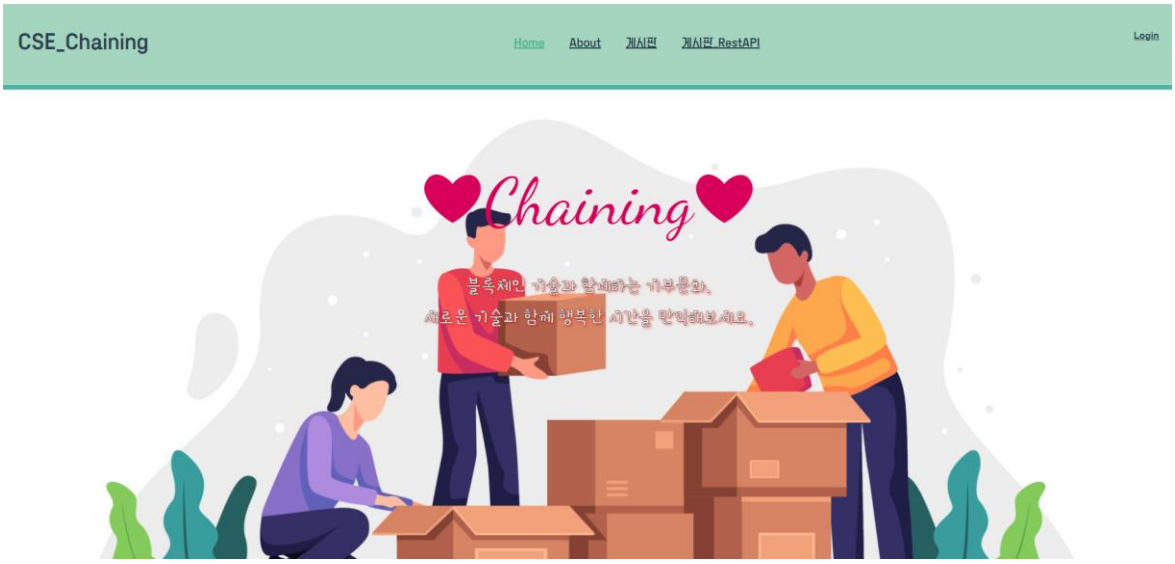
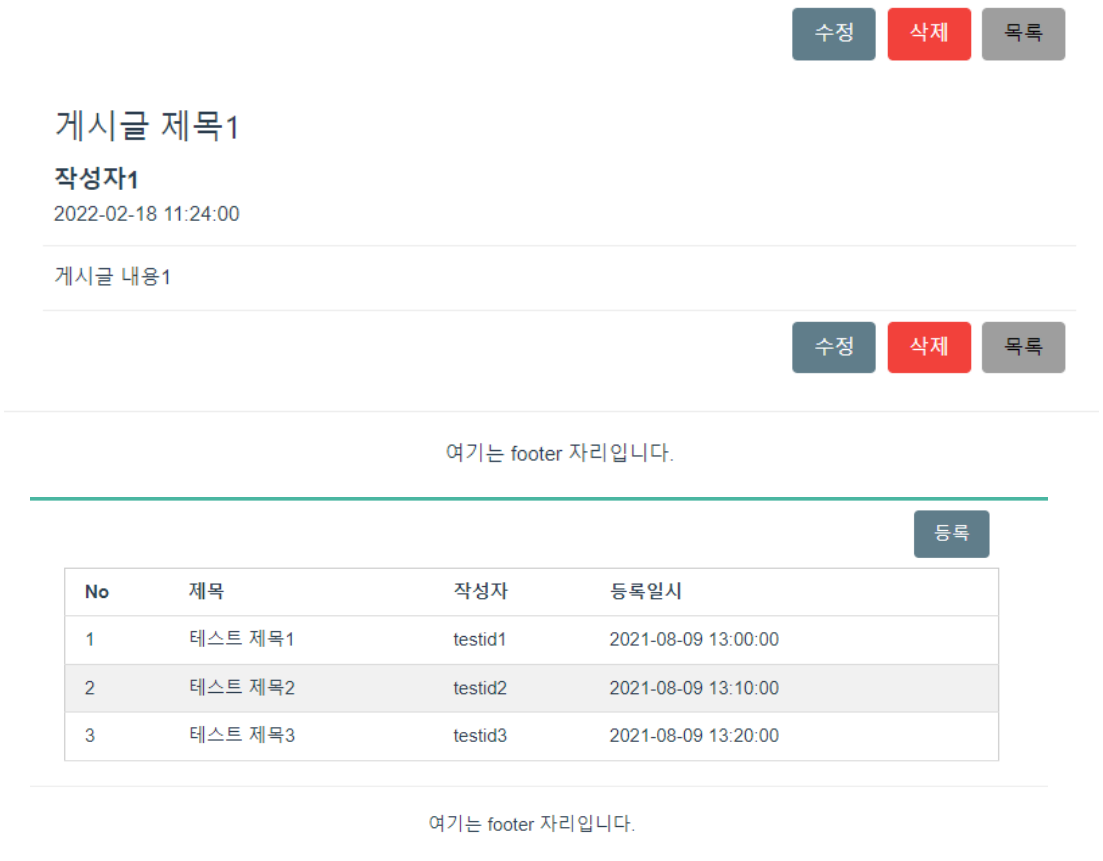


그림 7. 메인 화면



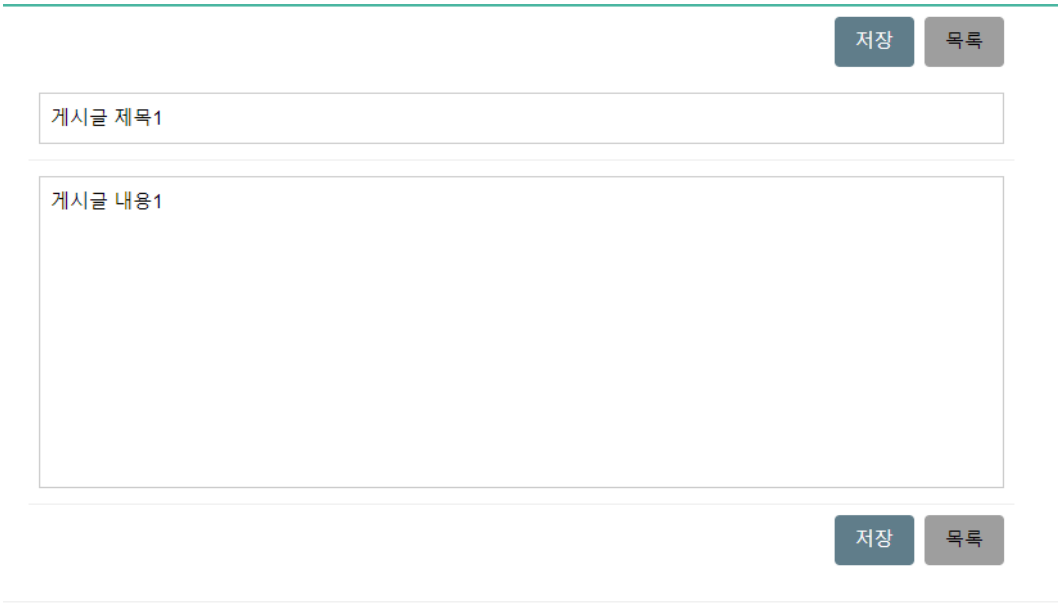


그림 8. 게시판 구현

- 현재 Jquery 작업 중
- 추후 블록체인과 연동하여 송금 기능 및 마이페이지 등 구현

3) Back-end

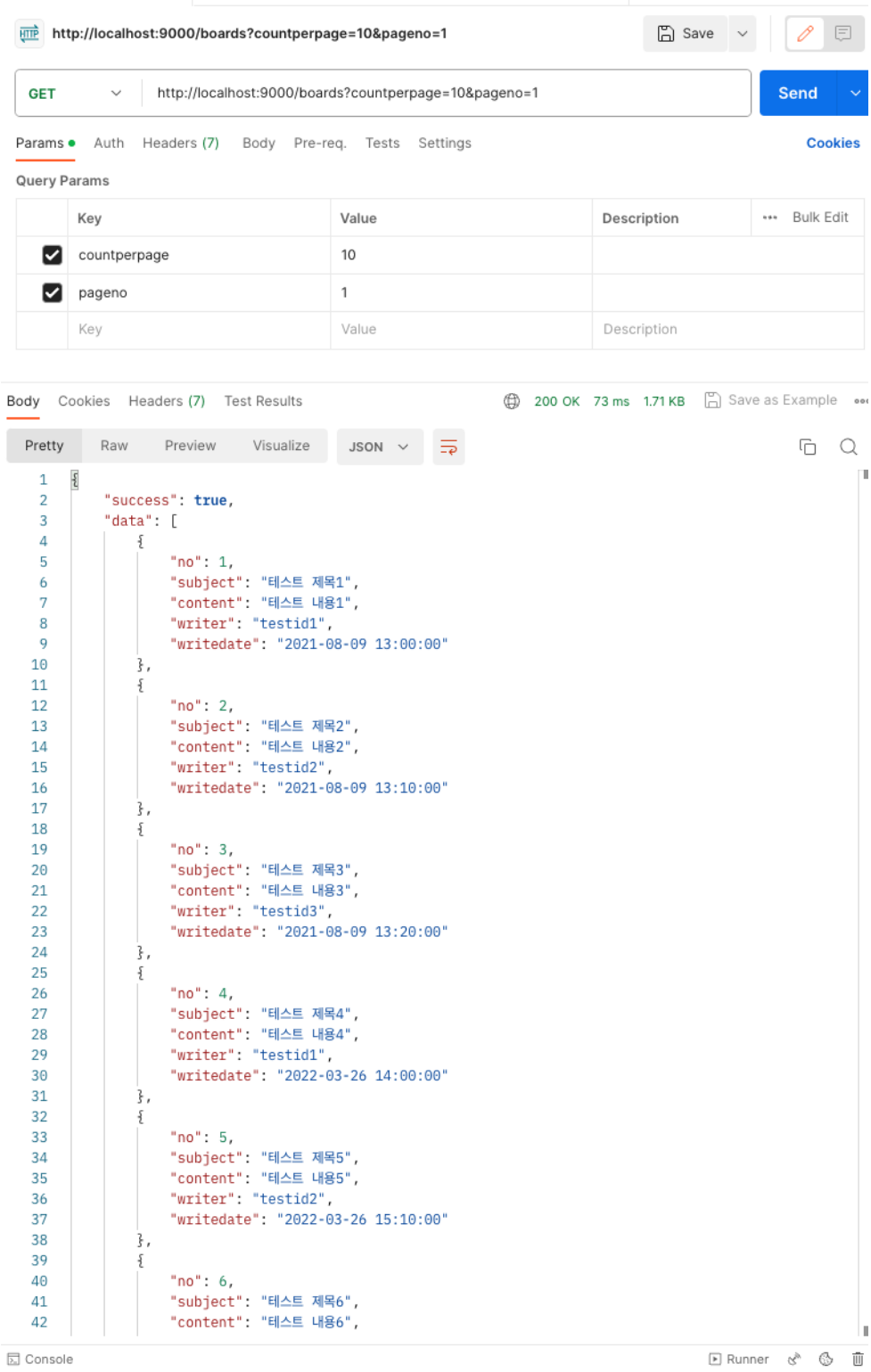


그림 9. REST API Pagination

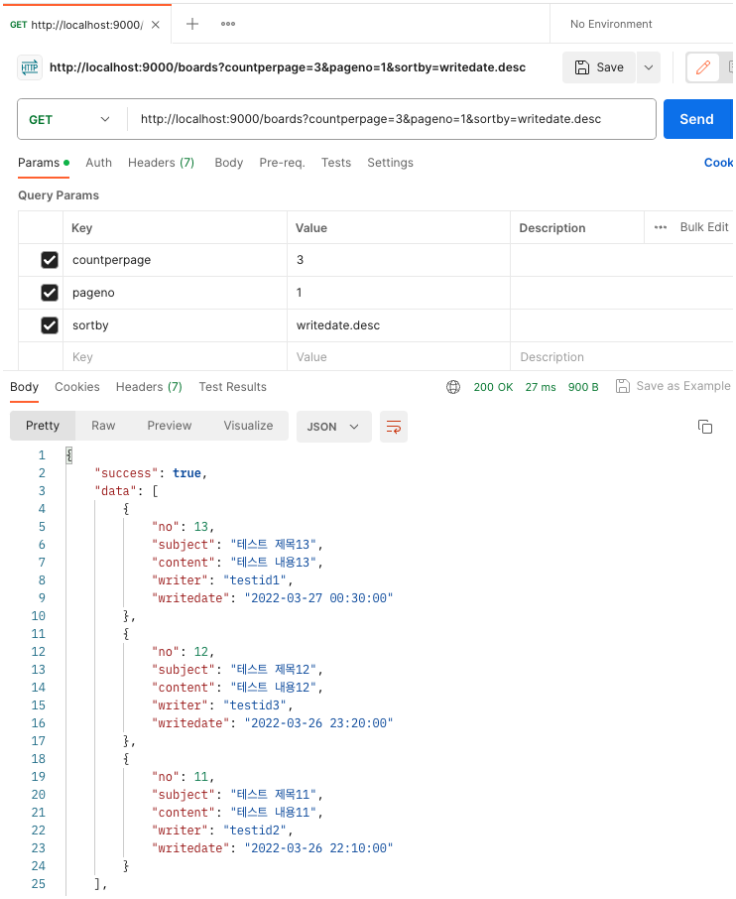


그림 10. REST API Sort

- 추후 로그인, 회원가입 등의 기능을 구현하며 데이터베이스 완전히 구축 예정

4. 향후 개발 일정

	8				9			
	1	2	3	4	1	2	3	4
웹사이트 개발								
데이터베이스 및 서버 구축								
블록체인 개발 및 연동								
테스트 및 디버깅								
최종보고서 및 발표심사 준비								